5H

УДК 519.713

Н.В. Ногина, И.С. Грунский
Институт информатики и искусственного интеллекта
ГВУЗ «Донецкий национальный технический университет»
Украина, 83050, г. Донецк, пр. Б. Хмельницкого, 84

Синтез регулярного выражения языка, порожденного помеченным графом, методом его локальной редукции

N.V. Nogina, I.S. Grunsky
Institute of Informatics and Artificial Intelligence
of Donetsk National Technical University
Ukraine, 83050, c. Donetsk, B. Khmelnitsky ave., 84

Synthesis of Regular Expression for Language Generated by a Labeled Graph by Means of its Local Reduction

Н.В. Ногіна, І.С. Грунський
Інститут інформатики і штучного інтелекту
ДВНЗ «Донецький національний технічний університет»
Україна, 83050, м. Донецьк, пр. Б. Хмельницького, 84

Синтез регулярного виразу мови, що породжена поміченим графом, методом його локальної редукції

Предлагается новый алгоритм анализа языков, порожденных графами с помеченными вершинами и дугами. Он позволяет находить алгебраическое выражение (в терминах соответствующей алгебры) таких языков. Алгоритм основан на локальной редукции графа, т.е. на последовательном исключении его вершин и дуг. Предложен порядок редукции, при котором исключение вершин проводится от финальной к начальной, а также упрощение графа в процессе редукции, что зачастую позволяет уменьшить объем вычислений.
Ключевые слова: помеченный граф, алгебра языка, регулярное выражение, локальная редукция графа.

New algorithm for analysis of languages generated by graphs with labeled vertices and transitions is proposed. It gives regular expression (in terms of the proper algebra) describing the language. The algorithm is based on a local reduction of the graph, that is the sequential exclusion of vertices and transitions. It is proposed a reduction procedure, in which removal starting at the final vertex to initial, and a simplification of the graph in the process of reduction, which often reduces the amount of computations.
Key Words: labeled graphs, algebra of language, regular expression, local reduction of the graph.

Запропоновано новий алгоритм аналізу мов, породжених графами з поміченими вершинами і дугами. Він дозволяє знаходити алгебраїчний вираз (в термінах відповідної алгебри) таких мов. Алгоритм засновано на локальній редукції графа, тобто на послідовному виключенні його вершин та дуг. Запропоновано порядок редукції, при якому видалення вершин проводиться від фінальної до початкової, а також спрощення графа в процесі редукції, що часто дозволяє зменшити обсяг обчислень.
Ключові слова: помічений граф, алгебра мови, регулярний вираз, локальна редукція графа.

Введение

В настоящее время актуальны задачи, связанные с анализом и синтезом языков, представимых в помеченных графах. Интерес к изучению таких графов вызван тем,

что существует ряд важных задач, которые естественным образом представляются в виде помеченных графов. Такие графы интенсивно изучаются при верификации программ [1] и планировании движения мобильного робота [2].

Свойства языков, представимых графами с отмеченными дугами, изучались С. Клини. Им предложена широкоизвестная алгебра, описывающая эти языки. В настоящее время известен ряд алгоритмов описания этих языков в алгебре Клини и построение графов по такому описанию [3]. Свойства языков, представимых графами с отмеченными вершинами, рассматривались в работах [4-7]. В них предложены алгебры, аналогичные алгебре Клини, для описания языков, порожденных такими графами, и алгоритмы перехода от алгебраических выражений, описывающих языки, к представляющим их графам и наоборот.

В данной работе рассматривается алгебра, описывающая языки, порожденные графами с помеченными вершинами и дугами, частным случаем которой является алгебра, описанная в [4-7]. Рассматривается задача построения алгебраического выражения языков, представимых в таких графах.

Целью данной работы является разработка нового алгоритма построения регулярного выражения по заданному помеченному графу.

Основные определения и обозначения

Помеченным графом назовем восьмерку $G=(Q,E,X,Y,\mu,\rho,q_0,F)$, где Q – конечное множество вершин, E – множество дуг, X – множество отметок вершин, Y – множество отметок дуг, $\mu:Q\to X$ – функция разметки вершин, $\rho:L\to Y$ – функция разметки дуг, q_0 – начальная вершина графа, F – множество финальных вершин. Путем в графе G будем называть конечную последовательность

$$l=q_1e_1q_2e_2...e_{k-1}.q_k,$$

где q_i – вершина, а e_i – дуга, началом которой является вершина q_i, а концом q_{i+1}. Отметка пути l – это последовательность отметок

$$w(l)=x_1y_1x_2y_2...y_{k-1}.x_k,$$

где $x_i=\mu(q_i)$, $y_i=\rho(e_i)$.

Успешным путем в помеченном графе назовем путь из начальной вершины в финальную. Языком $L(G)$, порождаемым графом G, назовем множество отметок всех успешных путей.

Пусть $Pre(q_i)$ – множество начальных вершин всех дуг, входящих в q_i, $Post(q_i)$ – множество конечных вершин всех дуг, исходящих из q_i.

Преходящей вершиной назовем вершину $q \neq q_0$, у которой $Pre(q)=\varnothing$, а висячей – $q \neq fin$, у которой $Post(q)=\varnothing$.

Пусть Z^+ – множество всех непустых слов вида $w=x_1y_1...y_{k-1}x_k$. Рассмотрим алгебру $\left\langle 2^{Z^+},\circ,\cup,\circledast,\varnothing,Z\right\rangle$, в которой операции на языках $L_1, L_2, L \subseteq Z^+$ определены следующим образом:

1) операция объединения: $L_1\cup L_2 = \{w \mid w\in L_1 \; или \; w\in L_2\}$;

2) операция сочленения (склеивания) слов: $L_1\circ L_2=\{w_1'xw_2' \mid$ если $w_1=w_1'x$, $w_2=xw_2'\}$;

3) операция итерации (зацикливания): $L^{\circledast} = \bigcup_{i=0}^{\infty} L^i$, где $L^0 = L_{нач} \circ L_{кон}$, причем $L_{нач}=\{x \mid xw' \in L, x \in X\}$, $L_{кон}=\{x \mid w'x \in L, x \in X\}$; $L^1=L$; $L^{n+1}=L^n \circ L$ для всех $n \geq 1$.

Регулярные выражения определим индуктивно:

1) пустое множество $\varnothing$ является регулярным выражением;

2) x, xyx' являются регулярными выражениями для всех символов $x, x' \in X$, $y \in Y$;

3) если p и q – регулярные выражения, то выражения $p \circ q$, $p \cup q$, $p^{\circledast}$ также являются регулярными.

Постановка задачи

В данной работе решается следующая задача. Дан конечный ориентированный связный помеченный граф. Требуется разработать алгоритм построения регулярного выражения языка, порождаемого этим графом.

Решение систем линейных уравнений

Очевидно, что по аналогии с работой [5] для любого помеченного графа G, порождающего язык $L(G)$, можно найти регулярное выражение R, для которого $L(R)=L(G)$. Для этого необходимо решить систему линейных уравнений вида (1)

$$\begin{cases} L_1 = M_{11} \circ L_1 \cup M_{12} \circ L_2 \cup \ldots \cup M_{1n} \circ L_n \cup V_1 \\ L_2 = M_{21} \circ L_1 \cup M_{22} \circ L_2 \cup \ldots \cup M_{2n} \circ L_n \cup V_2 \\ \ldots \\ L_n = M_{n1} \circ L_1 \cup M_{n2} \circ L_2 \cup \ldots \cup M_{nn} \circ L_n \cup V_n \end{cases}, \quad (1)$$

где L_i – язык, порождаемый i-й вершиной графа, т.е. множество отметок всех путей из вершины q_i в любую финальную; $M_{ij} = \mu(q_i)\rho(e_i)\mu(q_j) = x_i y_i x_j$; $V_i = \mu(q_i) = x_i$, если вершина q_i является финальной, и $V_i = \varnothing$ в противном случае.

В [5] было показано, что линейное уравнение вида $L_i = A \circ L_i \cup B$ имеет единственное решение

$$L_i = A^{\circledast} \circ B \cup B. \quad (2)$$

Для решения системы линейных уравнений (1) можно использовать метод многократного удаления неизвестных, подобный классическому методу Гаусса. Используя (2), можно удалить L_n из правой части последнего уравнения в системе (1). Полученное значение для L_n можно подставить в предпоследнее уравнение и получить выражение для L_{n-1}, которое зависит только от $L_1 \ldots L_{n-2}$. Продолжая такие подстановки для всех оставшихся в системе уравнений, можно получить значения всех L_i, которые будут регулярными выражениями в рассматриваемой алгебре.

Алгоритм построения регулярного выражения

Вход. Граф G с отмеченными вершинами, с начальной и финальными вершинами.

Выход. Регулярное выражение языка, порожденного исходным графом.

Шаг 1. Граф G превращается в граф с отмеченными дугами. Для этого отметки вершин стираются и в дугах (q_i, e_i, q_j) отметкой e_i становится $x_i y_i x_j$, где $x_k = \mu(q_k)$, где $k=i,j$, $y_i = \rho(e_i)$.

В список вершин вводится фиктивная конечная вершина *fin*, а в список дуг – дуга из каждой финальной вершины q_i в вершину *fin* с отметкой вершины q_i.

Шаг 2. Удаление преходящих и висячих вершин.

While в графе существует вершина $q \neq q_0$, у которой $Pre(q_i)=\varnothing$ **do**

Удаляем вершину q_i и все дуги, исходящие из нее;

While в графе существует вершина $q_i \neq fin$, у которой $Post(q_i) = \varnothing$ **do**

Удаляем вершину q_i и все дуги, входящие в нее;

Шаг 3. **If** в графе существует хоть одна петля или существуют вершины, не являющиеся начальными, из которых исходит хоть одна дуга, **then goto** *Шаг 4*

else goto *Шаг 7*;

*Шаг 4***.** Удаление кратных дуг и петель.

1. Удаляем кратные дуги, заменяя их одной дугой с отметкой, равной объединению отметок исходных дуг.

2. Удаляем все петли по следующему правилу.

Пусть в вершине q_i есть петля с отметкой *A*. Если из этой вершины нет дуги в другую вершину, то петля удаляется. В противном случае для всех дуг (q_i, e_i, q_j), где $i \neq j$, с отметкой дуги *B*, петля удаляется, а отметка *B* заменяется отметкой $A^{\circledast}\circ B \cup B$.

На шагах 5 – 6 происходит удаление одной вершины.

Шаг 5.

Выбираем $q_i \in Pre(fin)$;

$q := q_i$;

Шаг 6.

If $q \neq q_0$ **then** удаляем вершину q и все входящие и выходящие из нее дуги. Если при этом есть произвольный путь из некоторой вершины q_i в вершину q_k, через удаляемую вершину q, где $q_j \in Pre(q)$ и $q_k \in Post(q)$, то в граф добавляется дуга, содержащая пометку, равную склеиванию пометок удаляемых дуг данного пути;

goto *Шаг 2*;

else q_i равная q_0 не исключается;

выбираем $q_m \in Pre(q_0)$;

$q := q_m$;

goto *Шаг 6*;

*Шаг 7***.** Удаляем все вершины $q \neq q_0$ и $q \neq fin$ и все входящие в них дуги. Получим граф, состоящий из двух вершин: q_0 и *fin* и дуги, между ними с пометкой *R*, где *R* – это искомое регулярное выражение.

Анализ алгоритма

Из описания алгоритма следует, что алгоритм имеет следующую структуру: шаги 2 – 6 образуют внешний цикл алгоритма, шаги 5 – 6 образуют вложенный цикл алгоритма. При прохождении один раз внешнего цикла в графе удаляется по крайней мере одна вершина. Следовательно, внешний цикл выполняется $O(n)$ раз, где n – число вершин помеченного графа. Таким образом, алгоритм завершает свою работу через конечное число шагов. При выполнении одного прохода по внешнему циклу на шаге 4 в графе удаляются кратные дуги и петли. Таким образом, перед началом выполнения шага 5 в графе может быть не более $O(n^2)$ дуг. При выполнении шагов 5 – 6 алгоритма удаляется одна вершина графа, при этом в графе удаляется не более $O(n)$ дуг, и может быть добавлено не более $O((n-1)^2)$ новых дуг.

Из проведенных рассуждений следует, что если анализ одной дуги графа проводится в единицу времени, то алгоритм требует не более $O(nm)$ единиц времени, где m – количество дуг в исходном графе.

Докажем правильность получаемого решения, показав, что все операции на графах, применяемые в данном алгоритме, фактически соответствуют методу решения систем линейных уравнений рассматриваемой алгебры.

1. Операция удаления вершины соответствует подстановке одного уравнения в другое:

$$\begin{cases} L_{qj} = A \circ L_q \\ L_q = B \circ L_{qk} \end{cases} \rightarrow L_{qj} = A \circ B \circ L_{qk}.$$

2. Операция удаления кратных ребер обоснована дистрибутивным законом рассматриваемой алгебры:

$$L_{qi} = A \circ L_{qj} \cup B \circ L_{qj} = (A \cup B) \circ L_{qj}.$$

3. Операция удаления петли соответствует решению (2) линейного уравнения с одним неизвестным данной алгебры.

4. Удаление висячих вершин на Шаге 2 обосновано тем, что язык, порождаемый такой вершиной, пуст, поскольку из этих вершин вершина *fin* недостижима.

5. Очевидно, что удаление преходящих вершин и всех исходящих из них дуг на Шаге 2 возможно без изменения множества успешных путей в помеченном графе, поскольку данные вершины недостижимы из начальной.

Таким образом, все операции на графах, выполняемые предложенным алгоритмом, соответствуют методу решения системы линейных уравнений в данной алгебре. Следовательно, алгоритм корректен.

Алгоритм основан на локальной редукции графа, т.е. на последовательном исключении его вершин и дуг. Предложен порядок редукции, при котором исключение вершин проводится от финальной к начальной. Такой порядок позволяет не обрабатывать вершины, из которых *fin* не достижима, и они удаляются все сразу на шаге 7.

В алгоритме предусмотрено упрощение графа. Оно достигается, во-первых, путем последовательного удаления всех преходящих и висячих вершин (шаг 2) и, во-вторых, путем удаления всех петель и кратных дуг (шаг 3) перед каждым удалением вершины (шаг 5 – 6). Таким образом, это упрощение, по сути, является построением аналога трима [8, с. 23]. Порядок исключения вершин и приемы упрощения графа являются, на наш взгляд, весомыми достоинствами предложенного алгоритма.

Выводы

Предложенный алгоритм позволяет находить регулярное выражение языка, порожденного графом с помеченными вершинами и дугами.

В отличие от алгоритмов решения этой задачи, основанных на решении системы линейных уравнений [4], [5], в данном алгоритме учитывается структура графа, что зачастую позволяет уменьшить объем вычислений.

Литература

1. Годлевский А.Б. Предикатные преобразователи в контексте символьного моделирования транзиционных систем / А.Б. Годлевский // Кибернетика и системный анализ. – 2010. – № 4. – С. 91-99.
2. Dudek G. Computational principles of mobile robotic / G. Dudek, M. Jenkin. – Cambridge Univ. press, 2000. – 280 p.
3. Хопкрофт Дж. Введение в теорию автоматов, языков и вычислений / Хопкрофт Дж., Мотвани Р., Ульман Дж. – [2-е изд.]. – М. : Издательский дом «Вильямс», 2002. – 528 с.
4. Капитонова Ю.В. Математическая теория проектирования вычислительных систем / Ю.В. Капитонова, А.А. Летичевский. – М. : Наука, 1988. – 296 с.

5. Грунский И.С. Об алгебре языков, представимых графами с отмеченными вершинами / И.С. Грунский, Е.А. Пряничникова // Труды Ин-та прикл. математики и механики НАН Украины. – 2009. – Т. 18. – С. 37-46.
6. Grunsky I. Languages Representable by Vertex-labeled Graphs / I. Grunsky, O. Kurganskyy, I. Potapov // Proc. of the 30th International Symposium on Mathematical Foundations of Computer Science. – 2005. – V. 3618. – P. 435-446.
7. Grunsky I. On algebra of Languages Representable by Vertex-labeled Graphs / I. Grunsky, I. Potapov, E. Pryanichnikova // Theoretical Computer Science. – 2012. – V. 426-427.
8. Eilenberg S. Automata, Languages and Machines. Vol. A / Eilenberg S. – Academic Press, New York and London, 1974. – 469 p.

Literatura

1. Godlevskij A.B. Kibernetika i sistemnyj analiz. 2010. № 4. S. 91-99.
2. Dudek G. Computational principles of mobile robotic. Cambridge Univ. press. 2000. 280 p.
3. Hopkroft Dzh. Vvedenie v teoriju avtomatov, jazykov i vychislenij. M. : Izdatel'skij dom "Vil'jams". 2002. 528 s.
4. Kapitonova Ju.V. Matematicheskaja teorija proektirovanija vychislitel'nyh sistem. M. : Nauka. 1988. 296 s.
5. Grunsky I.S. Trudy In-ta prikl. matematiki i mehaniki NAN Ukrainy. 2009. T.18. S. 37-46.
6. Grunsky I. Languages Representable by Vertex-labeled Graphs. Proc. of the 30th International Symposium on Mathematical Foundations of Computer Science. 2005. V. 3618 P. 435-446.
7. Grunsky I. On algebra of Languages Representable by Vertex-labeled Graphs. Theoretical Computer Science. 2012. V. 426-427.
8. Eilenberg S. Automata, Languages and Machines. Vol A. Academic Press, New York and London. 1974. 469 p.

RESUME

N.V. Nogina, I.S. Grunsky

Synthesis of Regular Expression for Language Generated by a Labeled Graph by Means of its Local Reduction

In this paper, the authors study languages generated by labeled graphs in the case when both vertices and transitions are labeled. The case when words of generated language contain only labels of transitions is considered in the finite automata theory. The corresponding languages are known as regular or automata languages and intensely investigated. In this paper, the more complicated case is studied, i.e. when words contain labels of transitions and vertices. The generated languages can be used in a research of the behavioral properties of certain models and processes. These graphs are intensively studied in program verification and motion planning of mobile robots.

The authors consider algebra of languages generated by graphs with labeled vertices and transitions. The algorithm for analysis of languages generated by such graphs is proposed. The algorithm is based on the local graph reduction, i.e. on the sequential exclusion of its vertices and transitions.

An order of reduction in which the removal of vertices is performed from final to initial one and the simplification of the graph in the reduction process are proposed.

Unlike other algorithms for this problem based on the solving of the system of linear equations, in this algorithm the graph structure is taken into account that often reduces an amount of computation.

Статья поступила в редакцию 05.06.2012.